Advanced Image Processing - Histograms, Noise, Smoothing and Sharpening

Ing. Viktor Kocur
viktor.kocur@fmph.uniba.sk

DAI FMFI UK

29.10.2020

Histogram

imhist

imhist(I) - displays the histogram, in case the output is directed to a variable the histogram is not shown and the values are stored in the variable as a vector.

Exercise

Convert the zatisie.jpg to grayscale (rgb2gray function). In its histogram we can see three peaks. Change the image so that the values belonging to just one of the three peaks are fully white (maximum intensity).

Intensity processing

Gamma correction

Contrast in the image can be changed with the gamma correction: $i_{out} = A \cdot i^{\gamma}$, where i are the intensities of the pixels of the image. The intensity values are expected to be between 0 and 1!

Linear expansion

To linearnly expand the intensity we can use the following:

$$i_{out} = \frac{i - min(I)}{max(I) - min(I)},$$

where i are intensity values of the pixels and I is the set of all the intensities in the image. We expect the values to be between 0 and 1.

Histogram equalization

Equalization

Histogram equalization is a method to change the intensity values in the image in a way that results in the most balanced histogram (closest to the uniform distribution).

histeq

histeq(I) - returns the image after histogram equalization.

Exercise

For the image krajinka.jpg use all of the 3 methods to change the intensity values. Display both the image and the corresponding histograms.

Thresholding

imbinarize

imbinarize(I) - returns the binarized image using the Otsu method.

imbinarize

imbinarize(I, t) - returns the binarized image with the threshold t.

Exercise

Try to use thresholding on images coins.png, qr.jpg a zatisie.jpg.

Adaptive thresholding

imbinarize

imbinarize(I, 'adaptive') - returns binarized image using adaptive thresholding.

Exercise

Try adaptive thresholding for the images coins.png a qr.jpg.

Noise

How does noise occur?

Noise can be introduced in multiple stages of the image capture. This can occur on the sensor or during the transmission of the information.

Why do we model noise?

In reality it is impossible to avoid noise. It is therefore important to have a model of the noise so we can suppress it efficiently.

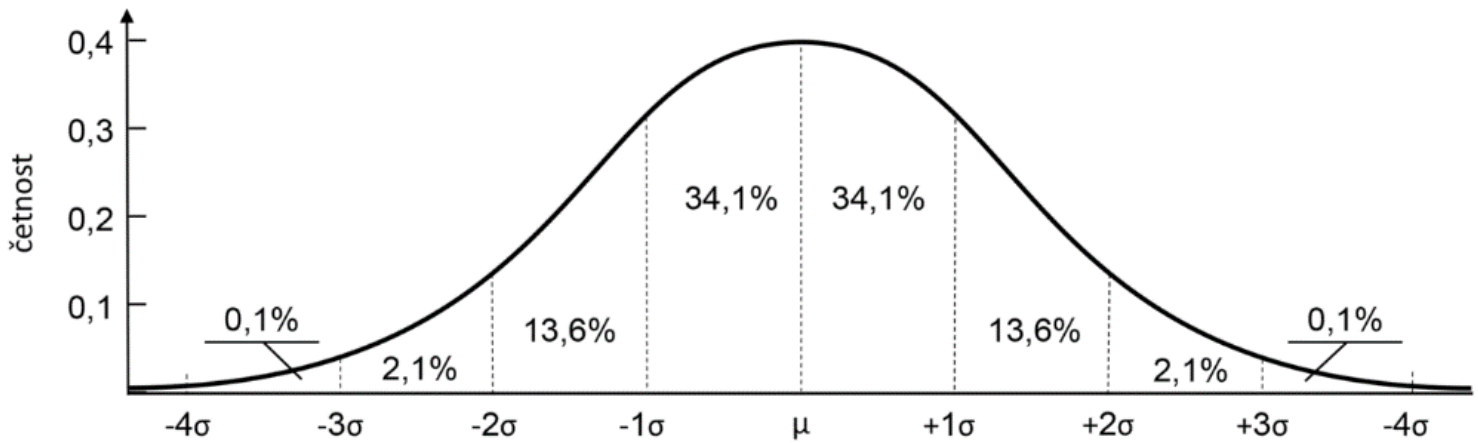
Additive Gaussian noise

Additivity

Noise is additive if $I = I_{orig} + S$.

Guassian character of the noise

$$P(S_{i,j} = x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (1)$$



Additive Gaussian noise in Matlab

randn

randn(sz) - returns matrix of size sz (e.g. sz = size(I)) with random elements from a normal distribution with $\sigma = 1$ a $\mu = 0$.

Exercise

Create a function zasum(I, sigma), which takes an image I (we can expect this to be in the double type and grayscale) and adds additive Gaussian noise with $\mu = 0$ and $\sigma = \sigma$. Make sure that the output is within the range 0 to 1. Test the function on an image with different values of σ.

Salt and Pepper

Salt and Pepper

Salt and pepper noise occurs when a pixel becomes completely dark or bright.

rand

rand(sz) - returns a matrix of size sz with a uniform distribution in the range from 0 to 1.

Exercise

Create a function okoren(I, p1, p2), which takes a grayscale image I (in double type) and adds salt and pepper noise such that with the probability of p1 we obtain a fully bright pixel and with the probability of p2 we obtain a fully dark pixel. Test the function with different parameters.

Smoothing

Why smoothing

When a human looks at a noisy image it is still possible to process and understand it. However we will be using various techniques that are sensitive to noise. It is therefore sometimes necessary to suppress the noise. This can be achieved by smoothing the image.

Convolution

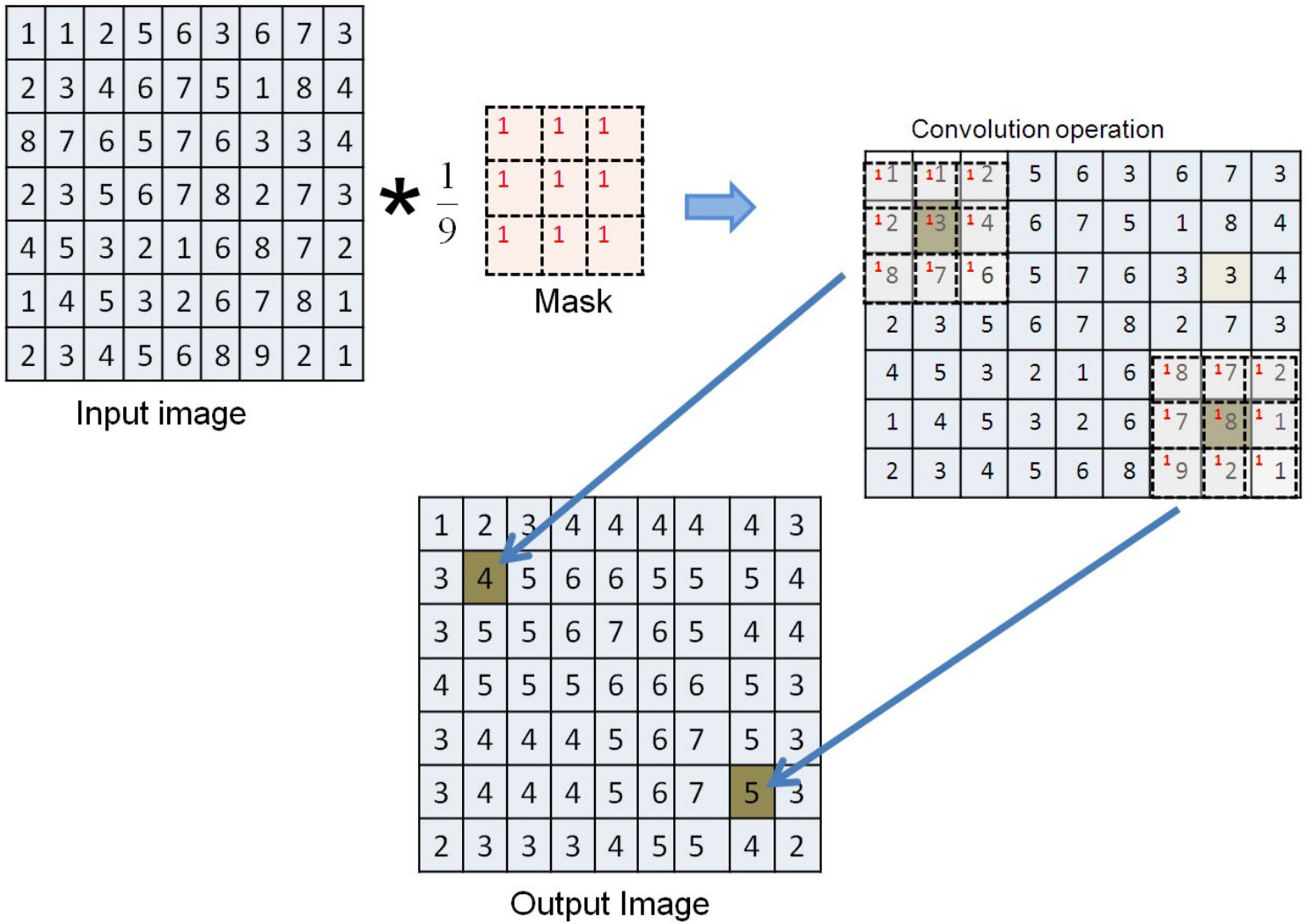
Convolution - Integral version - Real Numbers

$$J = I * M \iff J(\chi, \psi) = \int_{-\infty}^{\infty} \int_{\infty}^{\infty} I(x, y) M(\chi - x, \psi - y) dx dy$$

Convolution - Discrete version - Real Numbers

$$J = I * M \iff J(r, c) = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} I(u, v) M(r - u, c - v)$$

Convolution



Convolution - Matlab

conv2

conv2(A,B) - performs convolution of matrices A and B.

imfilter

imfilter(I,f) - applies a convolutional filter f on the image f

imfilter

imfilter(I,f, 'option') - option changes the size of the resulting image (e.g. 'same') by deciding what values are used when the convolution "looks" outside of the image (for example: 'replicate', 'symmetric'). It is possible to use multiple options at the same time.

Convolution - filters

Manual

Filters can be created manually e.g. averaging filter is ones(3)/9.

fspecial

fspecial('name', params) - returns a filters based on name and params.

fspecial

fspecial('average',hsize) - returns the averaging filter with size hsize
fspecial('gaussian',hsize,sigma) - returns a Gaussian filter

Convolution filters

Exercise

Display the Gaussian filters of various sizes and sigmas.

imgaussfilt

imgaussfilt(I, sigma) - filters the image with a Gaussian filter with a given sigma. This is more efficient than fspecial with imfilter.

Exercise

Add noise to an image with the method zasum. Try to suppress the noise with smoothing try this for the salt and pepper noise as well.

Median filtration

Median filtration

For salt and pepper noise the averaging method is not effective. A different approach is therefore necessary. Instead of averaging we will use the median value for each filtration window.

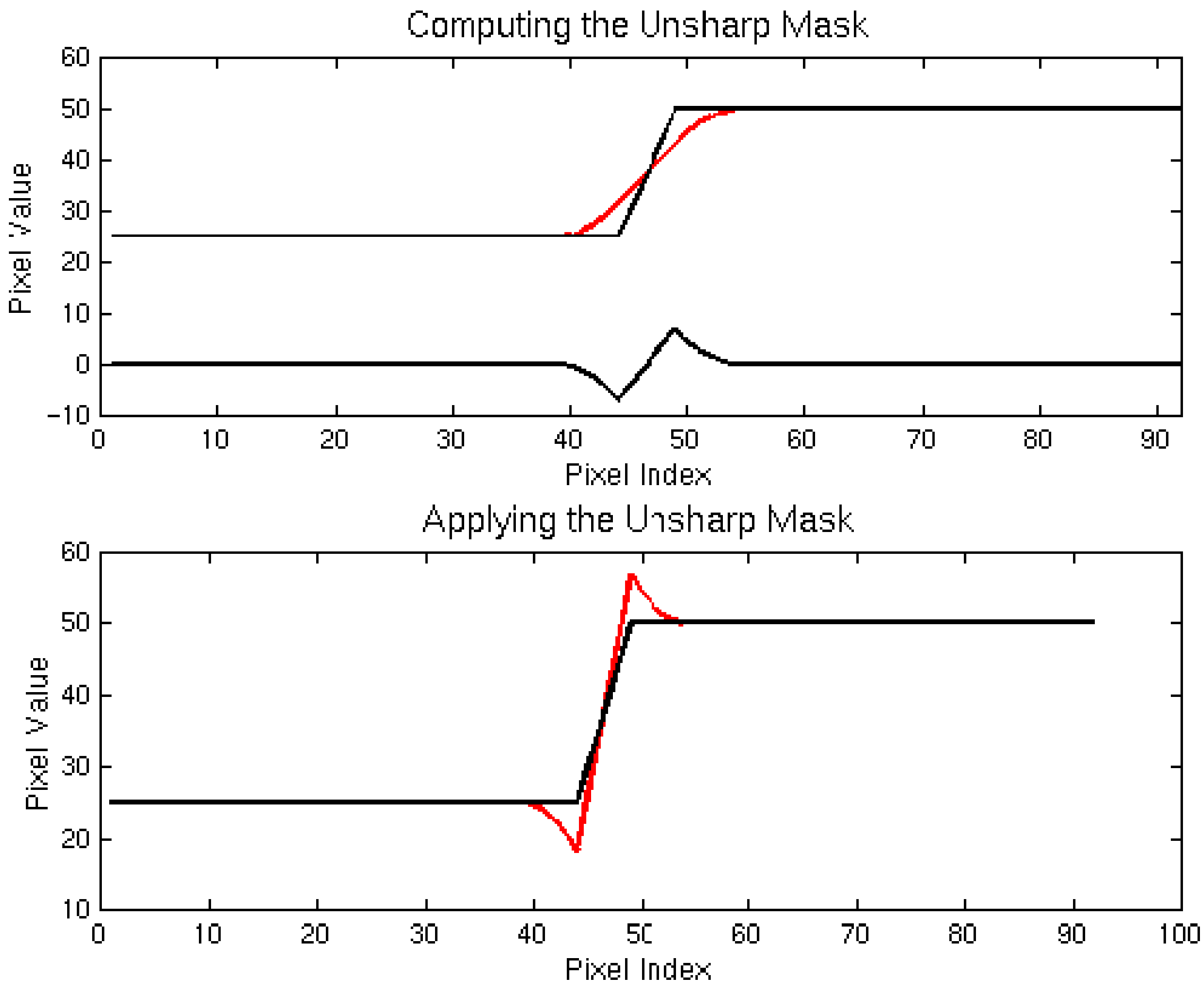
medfilt2

medfilt2(I, [m n]) - returns the image after median filtration with window size of m $\times$ n.

Exercise

Add Gaussian noise to an image and apply the median filtration. Try this also for the salt and pepper noise with various parameters.

Unsharp masking



Unsharp masking

Sharpening

We want to sharpen a blurred image. This task is similar to amplifying the edges.

Unsharp masking - princíp

$$I_{sharp} = I_{original} + p \cdot (I_{original} - I_{smooth})$$

Úloha

Create a function unsharp_mask(I,p,sigma), which applies unsharp masking with the parameter p with the use of addiive Gaussian noise with the parameter sigma. Apply this on the image blurred.pgm